Programming Assignment II
Sorting Algorithms

Due Date: 2022/10/13 (30)

1 Description of Assignment

Implement the following sorting algorithms and compare their performances.

1. Insertion sort
2. Selection sort
3. Quick sort
4. Merge sort
5. Heap sort

The first set of input data to be sorted should be randomly generated. Use the same random number generator and the same seed for each of the sorting algorithm. Then, if you still have time, try using other types of input data (for example, sorted and reversed data) to analyze the performances of the sorting algorithms.

Print out the CPU time required to sort n elements for each sorting algorithm, and include them in the report of your assignment. The number of elements to be sorted should be from $n = 10000$ to $n = 100000$, or even more. If the number of elements is too small, it may not be possible to distinguish the performances of the algorithm.

Before comparing the performances, make sure that all sorting algorithms are implemented correctly. Therefore, your program should provide debug mode to check the results of sorting. Do not include the time for checking the correctness of each sorting algorithm in the CPU time of that sorting algorithm.

Notes

The format of the report of the assignment does not need to be very formal, but it should be close to the format of a research technical report.

1. Title and author.
   Include assignment number, *your name*, *student number* and *email address* on the *first* page of your report.

2. Statement of the problem.
   A “formal” description of the problem in this assignment. In addition to the basic requirements specified in the assignment, emphasize the functions or features you implemented.

3. Main results.
   This section should include at least the following items.

   (a) Description of the design of your program and the data structures used in your program.

   (b) List of your program with comments. If your program is very long, list only the main parts of the program here and the entire program in an appendix. Additional comments can be added manually to explain the design of the program.

   (c) Outputs of the compilation and the executions of your program.

4. Conclusions
   Give a brief summary of what you did, and interesting thing you learned from this assignment.

Additional notes:

1. Turn in your report on or before due day.

2. The output of the program execution should indicate the correctness of your program. In other words, a set of comprehensive (but not necessarily exhaustive) annotated test data for the problem should be provided to show that your program is indeed correct. This can be done by carefully selecting a set of test data.

3. Print or write the report on A4 papers. Bind them together in the upper left corner.